A completion of some coxeter groups

Philippe Le Chenadec

IRIA

CENTRE DE ROCQUENCOURT

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105
78153 Le Chesnay Cedex
France
Tél.:(3) 954 90 20

Rapports de Recherche

N° 396

A COMPLETION OF SOME COXETER GROUPS

Philippe LE CHENADEC

Mai 1985

A Completion of some Coxeter Groups

Philippe Le Chenadec
INRIA, Domaine de Voluceau
Rocquencourt B.P. 105
78153 Le Chesnay Cedex, France

ABSTRACT

A set of identities, possibly infinite, in Coxeter groups without commuting pairs of generators is proved confluent and well-founded. An efficient algorithm for the word problem is extracted from the identities. Finally examples are given including some crystallographic groups. This case study outlines interesting features of the completion procedure, namely its ability to handle both families of groups and infinite sets of relations, from which follows the possibility to design efficient word algorithms.

RESUME

Un ensemble de règles de récriture, éventuellement infini, dans les groupes de Coxeter dont les générateurs ne commutent pas est montré confluent et noethérien. Un algorithme efficace pour le problème du mot dans ces groupes est déduit de ces règles. Quelques exemples sont présentés incluant des groupes cristallographiques. Cette étude met en évidence la capacité de traiter des familles de groupes et/ou des ensembles infinis d'égalités par la procédure de Knth et Bendix. De cette capacité découle la possibilité d'obtenir des algorithmes efficaces pour le problème du mot.

KEYWORDS

Finitely Presented Groups, Coxeter Groups, Word Problem, Confluence, Noetherianity, Knuth-Bendix Procedure.

MOTS-CLEFS

Goupes Finiment Présentés Groupes de Coxeter, Problème du Mot, Confluence, Noethérien, Procédure de Knuth et Bendix.

A Completion of some Coxeter Groups

Philippe Le Chenadec
INRIA, Domaine de Voluceau
Rocquencourt B.P. 105
78153 Le Chesnay Cedex, France

ABSTRACT

A set of identities in Coxeter groups without commuting pairs of generators is proved confluent and well-founded. This set, possibly infinite, was intuited by experiments with the Knuth-Bendix completion procedure. An efficient algorithm for the word problem is extracted from the identities. Finally examples are given including some crystallographic groups. This case study outlines interesting features of the completion procedure, namely its ability to handle both ***families*** of groups and ***infinite*** sets of relations, from which follows the possibility to design efficient word algorithms.

1. INTRODUCTION

In [7], the word problem for Coxeter groups is proved decidable with an algorithm enumerating the finite set of derivatives of a word under a reduction relation. This reduction is confluent but not noetherian (well-founded). We strengthen it into a noetherian relation found with the Knuth-Bendix completion procedure [4]. However, the new relation does not hold in the case of groups having commuting pairs of generators.

From this example, we can argue that the completion procedure is a good tool in the study of finitely presented groups. By opposition to the Todd-Coxeter coset enumeration [3], it applies to infinite groups as well as to families of groups. Moreover, in some cases where the procedure loops, the infinite set of rules may be finitely presented (see example $G3$). These facts follow from the observation that the rules usually present a simple structure, which in turn is the starting point for the design of word problem algorithms.

Two drawbacks are encountered. First, the partial commutativity of a presentation leads to a failure, this situation is very closed to the general completion procedure where specialized algorithms must be developed to handle both commutativity and associativity [6]. Second, theorem 3 develops a hand-made meta-completion accepting parametrized presentations as input. Clearly, this involves an extension of the classical completion, also needed by the completion of other group families (surface groups, polyhedral groups and symmetric groups, cf. [6]).

For finite groups, we may briefly compare the Todd-Coxeter coset enumeration [3] and the Knuth-Bendix completion. In terms of Cayley diagrams, the coset enumeration computes a representation of the Cayley graph, while the completion, by a computation on its cycles, determines a unique path between two vertices. It is therefore obvious that, as quoted by R.H. Gilman [8], the coset enumeration is generally more efficient (for the group E_6, a referee kindly reports that the Canberra implementation of coset enumeration produced a full coset table in less than three minutes, while we could not complete this group). Therefore, these two algorithms appear to be complementary, one being used for isolated groups, the other for parametrized families.

This section defines the Coxeter groups [1] and presents the completion procedure for groups [6]. The second section details the structure of the complete presentation, and proves its correctness. The third one extracts a reduction algorithm computing irreducible words. The last section is devoted to some examples.

1.1. Definition of Coxeter Groups

Let I be a finite set of **n** generators. A Coxeter matrix on I is a function $M: I\times I \longrightarrow \mathbb{N}\cup\{\infty\}$ such that for all i,j in I, $M(i,i)=1$ and $M(i,j)=M(j,i)\geq 2$ if $i\neq j$. The value $M(i,j)$ will be denoted by m_{ij}. The Coxeter Group $C(M)$ is the quotient of the free group $F(I)$ on I by the normal subgroup $N(M)$ generated by $(ij)^{m_{ij}}$ when $m_{ij}\neq\infty$: $C(M)$ is defined by the presentation (I,E) where E is the set of equations $(ij)^{m_{ij}}=1$, $m_{ij}\neq\infty$. As $m_{ii}=1$ implies $i^{-1}=i$, we may represent the elements of $C(M)$ by words from the free monoid $L(I)$ on I. If the words w and w' define the same element of $C(M)$ we write $w=_M w'$. Syntactic equality (equality in $L(I)$) is noted $w=w'$. The length of a word w is noted $|w|$ ($|1|=0$ and $|iw|=1+|w|$ if $i\in I$).

Throughout the paper, $[ij]^k$ will denote the product $ijij\cdots$ of k generators alternatively equal to i and j; α will denote $[ij]^{m_{ij}-1}$. The generators i and j will be denoted by $f(\alpha)$ and $s(\alpha)$ respectively, and $l(\alpha)$ is the last generator of α, equal to i (resp. j) when m_{ij} is even (resp. odd). To α is associated the word $\overline{\alpha}=[ji]^{m_{ij}-1}$. Finally, m_{ij} will be abbreviated in m_α. The same definitions stand for $\beta=[ij]^{m_{ij}}$ and $\gamma=[ij]^{m_{ij}-2}$.

A first solution to the word problem is given by a theorem due to J. Tits (theorem 1, p. 93 of [1]). If the generators from I are interpreted by the following linear transformations of a real vector space with basis $e_1,\ldots,e_{\mathbf{n}}$:

$$s_i : e_j \longrightarrow e_j - 2(\cos\frac{\pi}{m_{ij}})e_i$$

then a word $w=i_1\cdots i_k$ from $L(I)$ represents the unit element in $C(M)$ iff $s_{i_1}\cdots s_{i_k}(\sum_{j=1}^{\mathbf{n}} e_j)=\sum_{j=1}^{\mathbf{n}} e_j$. As noted by J. Tits [7], this solution is not efficient.

A second solution was proposed in [7] based upon a reduction in $L(I)$ defined by the following replacement rules:

$wiiw' \longrightarrow ww'$, $i\in I$, $w,w'\in L(I)$.

$w\beta w' \longrightarrow w\overline{\beta}w'$, $w,w'\in L(I)$.

The confluence of this system is proved via the linear representation of Coxeter groups. As the reduction is not length increasing, the enumeration of the words reduced from a given one w halts, and we can check whether or not $w=_M 1$. The Knuth-Bendix completion may be used to significantly improve this algorithm, we now succinctly present this procedure in the framework of group theory.

1.2. Completion of a Group

Let $>_I$ be a total ordering on I. The associated lexicographic ordering $\gg_I$ on $L(I)$ is defined by:

$w \gg_I w'$ iff either $|w|>|w'|$
or $|w|=|w'|$ and $w=uiv$, $w'=ujv'$, $u,v,v'\in L(I), i,j\in I$ with $i>_I j$.

The lexicographic ordering is total, noetherian and compatible with the structure of $L(I)$: $w \gg_I w'$ implies $uwv \gg_I uw'v$ for all words u and v. When no confusion is possible, we simply denote a lexicographic ordering by $>$.

Let I^{-1} be the set of generators' inverses, denoted by i^{-1}, we put $i^{-1^{-1}}=i$ for all i^{-1} in I^{-1}. The group completion is defined on $\Sigma(I)=L(I\cup I^{-1})$. But we first need some technical definitions.

Rules

A rule $u \longrightarrow v$ is a pair of words (u,v) on $\Sigma(I)$ such that $u \gg_{I\cup I^{-1}} v$. A set R of rules defines a relation on $\Sigma(I)$, noted $\underset{R}{\Longrightarrow}$: $wuw' \underset{R}{\Longrightarrow} wvw'$ iff $u \longrightarrow v \in R$. As $u \gg_{I\cup I^{-1}} v$ and $\gg_{I\cup I^{-1}}$ is compatible with word concatenation, the reduction $\underset{R}{\Longrightarrow}$ is noetherian. The reflexive-transitive closure of $\underset{R}{\Longrightarrow}$ is noted $\overset{*}{\underset{R}{\Longrightarrow}}$.

Superpositions

Given two rules $u \longrightarrow v$ and $u' \longrightarrow v'$ in R such that there exist three words a,b,c with $b\neq 1$, $u=ab$ and $u'=bc$, we say that the former *superposes* on the latter with matching word b. The word abc reduces on both $p=av'$ and $q=vc$. The pair (p,q) is a *critical pair* between the two rules, abc is a *least common instance* of the two rules. To each rule $u \longrightarrow v$ we associate three *normal pairs* (iu_1,vj^{-1}), $(u_1j,i^{-1}v)$ and (u^{-1},v^{-1}), where $u=iu_1j$, $i,j\in I\cup I^{-1}$ and if $w=i_1\cdots i_n$, $n\in\mathbb{N}$, $i_j\in I\cup I^{-1}$, $1\leq j\leq n$, then $w^{-1}=_{def} i_n^{-1}\cdots i_1^{-1}$.

Confluence

A relation $\underset{R}{\Longrightarrow}$ is confluent iff

$$\forall w,u,v\in\Sigma(I)\ w\overset{*}{\underset{R}{\Longrightarrow}}u \text{ and } w\overset{*}{\underset{R}{\Longrightarrow}}v \Rightarrow \exists w'\in\Sigma(I)\ u\overset{*}{\underset{R}{\Longrightarrow}}w' \text{ and } v\overset{*}{\underset{R}{\Longrightarrow}}w'.$$

For instance, if there exists such a word w' for a critical or normal pair (p,q), this pair is said to be confluent.

The theorem of Knuth and Bendix [4] then defines a completion procedure. In group theory, this theorem can be stated as follows [6]:

Theorem 1

Given a set of rules R defining a noetherian relation $\underset{R}{\Longrightarrow}$ on $\Sigma(I)$, then the relation $\underset{R}{\Longrightarrow}$ is confluent iff all critical and normal pairs of R are confluent. Given two words in $\Sigma(I)$, they represent the same element in the group presented by (I,R) iff their irreducible forms are equal.

The set $F_I=\{(ii^{-1},1)\,|\,i\in I\cup I^{-1}\}$ defines a noetherian and confluent relation computing the usual reduced words for $F(I)$ from $\Sigma(I)$. Given a finite presentation(I,E) of a group, the Knuth-Bendix completion starts with a set of rules equal to F_I. Then it increases this set by orientating the equations from E into rules according to a given lexicographic ordering. It generates critical pairs by superposition. When reduced by the current set of rules, the members of critical and normal pairs may be syntactically distinct, the reduced words are then orientated into a new rule. The completion halts as soon as reduced forms of all critical and normal pairs are syntactically equal. The final set of rules is therefore confluent from theorem 1. But the algorithm may loop for ever. In such cases, the infinite set of rules computed is confluent, although it may be non-recursive [5]. A major point for both efficiency and proofs is to keep the rules interreduced.

2. COMPLETION OF A COXETER GROUP

We now detail the completion of a Coxeter group defined by a matrix $M=(m_{ij})$, under a given lexicographic ordering. Together with a constant set of rules, the completion generates new rules sharing a common structure described by a single meta-rule. The complete proof of correctness of the set of rules requires three steps:

1) The reduction is noetherian,

2) Every rule is a consequence of the definition of the group,

3) All critical and normal pairs are solved.

The first point follows from our choice of the lexicographic ordering. Also, we prove that the meta-rule is a consequence of the definitions. Then, we will check that all critical and normal pairs are solved or generate an instance of the meta-rule.

2.1. The meta-rule

First, we restrict ourselves to matrices having no entry equal to 2. The completion begins by a rearrangement of the given presentation:

Lemma 2

Given a Coxeter matrix M on the set I totally ordered by $>$, the completion generates the two following sets of rules:

$$R_I = \{ i^{-1} \longrightarrow i,\ ii \longrightarrow 1 \mid i \in I \} \text{ and } S_I = \{ \beta \longrightarrow \bar{\beta} \mid f(\beta) > s(\beta) \}.$$

R_I is generated by the defining relations $ii=1$ and their normal pairs for all i in I. We have $ii>1$, the pairs are (i^{-1},i) and $(i^{-1}i^{-1},1)$. Putting $i^{-1}>i$, the rule $i^{-1} \longrightarrow i$ is generated, under which the second pair is confluent.

S_I is generated by a sequence of normal pairs from the defining rules. If m_{ij} is even, this rule is $\beta\beta \longrightarrow 1$. It generates $\beta\alpha \longrightarrow l(\beta)$. The first rule is redundant and deleted as $\beta\beta$ reduces to $l(\beta)l(\beta)$, then to 1, by the new rule and R_I. This sequence of operations loops until a pair $(\beta,\bar{\beta})$ is created. Then, a rule of type S_I is produced . The case m_{ij} odd is similar ∎

From R_I we can restrict to words in $L(I)$. These rules are length decreasing, while those from S_I shift generators i and j (in [6], this process which balances a presentation is called symmetrization, $R_I \cup S_I$ is a symmetrized presentation of a Coxeter group). Note that a word α is both a left member suffix (resp. prefix) and a right member prefix (resp. suffix) of a rule in S_I when $f(\alpha)<s(\alpha)$ (resp. $>$), and that both $\alpha\alpha$ and $\alpha\bar{\alpha}$ reduce on 1.

Theorem 3

Let M be a Coxeter matrix on the set I totally ordered by $>$. If $m_{ij} \neq 2$, $i,j \in I$, then the completion procedure generates the set of rules $R_I \cup T_I$, where T_I consists of all rules of the form

$$\alpha_1 \cdots \alpha_n l(\overline{\alpha_n}) \longrightarrow s(\alpha_1)\alpha_1 \cdots \alpha_n \tag{T}$$

where n is a positive integer, and for all p such that $1<p<n$:

$$f(\alpha_1)>s(\alpha_1),\ s(\alpha_p)>f(\alpha_p),\ f(\alpha_{p+1}) \neq l(\alpha_p),\ s(\alpha_{p+1})=l(\overline{\alpha_p}) \tag{C}$$

A R-rule (resp. S, T) means a rule from R_I (resp. S_I, T_I). By C, both words $\beta_n=\alpha_n l(\overline{\alpha_n})$ when $n>1$ and $\overline{\beta_1}=s(\alpha_1)\alpha_1$ are right members of a S-rule. Taking

$n=1$, we have $S_I \subset T_I$, and from lemma 2 the completion generates R_I and S_I.

The first step is to prove that every instance of the meta-rule is a consequence of the definitions. This is achieved by an induction on n, lemma 2 claims the result for $n=1$. And we have the following identities:

$$\begin{aligned} & \alpha_1 \cdots \alpha_{n-1}\alpha_n l(\overline{\alpha_n}) \\ =_{\mu}\ & \alpha_1 \cdots \alpha_{n-1}s(\alpha_n)\alpha_n \quad \text{by } \beta_n =_{\mu} \overline{\beta_n} \\ =\ & \alpha_1 \cdots \beta_{n-1}\alpha_n \quad \text{by C and } \beta_{n-1}=\alpha_{n-1}l(\overline{\alpha_{n-1}}). \\ =\ & s(\alpha_1)\alpha_1 \cdots \alpha_{n-1}\alpha_n \quad \text{by induction hypothesis.} \end{aligned}$$

Thus, the T-rules are consequences of the definitions. The main point is to prove that all normal and critical pairs are solved. This is checked by an analysis of all superpositions. The analysis will also detail how are produced the T-rules. We need an assumption on the sequence of superpositions:

Assumption: Any T-rule is first superposed with S-rules, then on other T-rules.

By induction on the number of α's in a T-rule left member, we show that the completion effectively generates these pairs and only these ones. The induction hypothesis states that the completion limited to the critical and normal pairs of $R_n = R_I \cup \{ u \longrightarrow v \mid u \longrightarrow v \in T_I,\ u$ has at most n α's$\}$ generates R_{n+1}. We distinguish four superpositions:

i) The superpositions with matching equals to a single generator.

ii) The superpositions with R-rules.

iii) The superpositions with matching length greater than one.

iv) The normal pairs.

The following table will help the reader to keep trace of identities between the generators:

	m_α even	m_α odd
α	iji	ijij
$\overline{\alpha}$	jij	jiji
β	ijij	ijiji
$\overline{\beta}$	jiji	jijij

Case i.1 Let r be a T-rule $\alpha_1 \cdots \beta_n \longrightarrow \overline{\beta_1} \cdots \alpha_n$. We superpose this rule on a S-one $\overline{\beta_{n+1}} \longrightarrow \beta_{n+1}$, the matching subword is $l(\overline{\alpha_n}) = f(\overline{\beta_{n+1}})$.

We have two subcases, first $l(\alpha_n) \neq s(\overline{\beta_{n+1}})$. This is the only case creating rules. The least common instance reduces on both:

$$\begin{aligned} \alpha_1 \cdots \alpha_n l(\overline{\alpha_n})\alpha_{n+1} & \longrightarrow s(\alpha_1)\alpha_1 \cdots \alpha_n\alpha_{n+1} = A \quad \text{by rule } r. \\ & \longrightarrow \alpha_1 \cdots \alpha_n\beta_{n+1} \quad \text{by } \overline{\beta_{n+1}} \longrightarrow \beta_{n+1} \\ & = \alpha_1 \cdots \alpha_n\alpha_{n+1}l(\overline{\alpha_{n+1}}) = B \end{aligned}$$

The critical pair (A,B) has the expected structure, the link condition being satisfied between α_n and α_{n+1}. But such a pair must be non-confluent in order to create a rule. We prove its irreducibility under the current set of rules R_n. Both words $s(\alpha_1)\alpha_1 \cdots \alpha_n$ and $\alpha_{n+1}l(\overline{\alpha_{n+1}})$ are irreducible as right members. Condition C being satisfied, no R-reduction is possible. Thus, any S or T-reduction must reduce a suffix of α_n and a prefix of α_{n+1}, but the concatenation of these words lôoks like $\cdots ij.ki \cdots$ with $k \neq i,j$. We cannot reduce generator i as, for A, we do not have the last generator $l(\overline{\alpha_{n+1}})$, and for B we must use a T-rule of $n+1$ components by inequalities C, which is impossible by induction hypothesis. The only possibility is a reduction by the S-rule $jk \longrightarrow kj$ if it exists. But this is impossible by the assumption $m_{jk} \neq 2$. Reciprocally, if the pair $p=(\alpha_1 \cdots \beta_{n+1}, \overline{\beta_1} \cdots \alpha_{n+1})$ is an instance of the meta-rule, then both rules $\overline{\beta_{n+1}} \longrightarrow \beta_n$ and $\alpha_1 \cdots \beta_n \longrightarrow \overline{\beta_1} \cdots \alpha_n$ are generated by induction hypothesis. As C holds for p, $f(\overline{\beta_{n+1}})=s(\alpha_{n+1})=l(\overline{\alpha_n})$. The superposition of the two rules with matching $f(\beta_{n+1})$ creates the rule p as detailed just above. Thus, at least all T-rules are effectively generated. We now check that no other ones are generated.

In the subcase where $l(\alpha_n)=s(\overline{\beta_{n+1}})$, m_{α_n} is even and $\alpha_n=\alpha_{n+1}$. Thus, $\alpha_n \alpha_{n+1} \xrightarrow[R]{*} 1$:

$$\begin{aligned}
A &\xrightarrow[R]{*} s(\alpha_1)\alpha_1 \cdots \alpha_{n-1} \\
B &\xrightarrow[R]{*} \alpha_1 \cdots \alpha_{n-1} l(\overline{\alpha_{n+1}}) \\
&= \alpha_1 \cdots \alpha_{n-1} l(\overline{\alpha_{n-1}}) \quad \text{by } l(\overline{\alpha_{n+1}})=l(\overline{\alpha_n})=s(\alpha_n) \text{ and C}. \\
&\xrightarrow[T]{} s(\alpha_1)\alpha_1 \cdots \alpha_{n-1} = A \quad \text{by induction hypothesis}.
\end{aligned}$$

The critical pair is confluent in this subcase.

Case i.2 The left part of the least common instance comes from a S-rule. With $f(\alpha_0)>s(\alpha_0)$, $f(\alpha_1)>s(\alpha_1)$ and $l(\overline{\alpha_0})=f(\alpha_1)$, we have:

$$\begin{aligned}
\alpha_0\alpha_1 \cdots \alpha_{n-1}\beta_n &\longrightarrow \overline{\beta_0}\overline{\gamma_1}\overline{\alpha_2} \cdots \alpha_{n-1}\beta_n = A \quad \text{by } \beta_0 \longrightarrow \overline{\beta_0}. \\
&\longrightarrow \alpha_0 s(\alpha_1)\alpha_1 \cdots \alpha_n \quad \text{by right reduction}. \\
&= \alpha_0\overline{\alpha_1}l(\alpha_1)\alpha_2 \cdots \alpha_n = B
\end{aligned}$$

We then have two subcases. First $f(\overline{\alpha_1})=s(\alpha_1)\neq l(\alpha_0)$:

$$\begin{aligned}
B &\xrightarrow[T]{} s(\alpha_0)\alpha_0\overline{\alpha_1}\alpha_2 \cdots \alpha_n \quad \text{by induction hypothesis (C holds for } \alpha_0 \text{ and } \overline{\alpha_1}). \\
&= \overline{\beta_0}\overline{\gamma_1}l(\overline{\alpha_1})\alpha_2 \cdots \alpha_n \\
&= \overline{\beta_0}\overline{\gamma_1}\overline{\beta_2}\alpha_3 \cdots \alpha_n \quad \text{by } s(\alpha_2)=l(\overline{\alpha_1}). \\
&\longrightarrow \overline{\beta_0}\overline{\gamma_1}\alpha_2 l(\overline{\alpha_2})\alpha_3 \cdots \alpha_n \quad \text{by } \overline{\beta_2} \longrightarrow \beta_2 \text{ as } s(\alpha_2)>f(\alpha_2). \\
&\cdots \\
&\longrightarrow \beta_0\overline{\gamma_1}\alpha_2 \cdots \alpha_n l(\overline{\alpha_n}) = A.
\end{aligned}$$

The reduction $l(\overline{\alpha_1})\alpha_2 \cdots \alpha_n \xrightarrow[S]{*} \alpha_2 \cdots \alpha_n l(\overline{\alpha_n})$ will be used again and called a shift/reduce sequence.

Second, $f(\overline{\alpha_1})=s(\alpha_1)=l(\alpha_0)$. This equality implies $\alpha_0=\overline{\alpha_1}$. Thus $\alpha_0\overline{\alpha_1} \xrightarrow[R]{*} 1$ and $\overline{\beta_0}\overline{\gamma_1} \xrightarrow[R]{*} l(\alpha_1)l(\overline{\alpha_1})$.

$$\begin{aligned}
A &\xrightarrow[R]{*} l(\alpha_1)l(\overline{\alpha_1})\alpha_2 \cdots \alpha_n l(\overline{\alpha_n}) \\
&\xrightarrow[S]{*} l(\alpha_1)\alpha_2 \cdots \alpha_n l(\overline{\alpha_n})l(\overline{\alpha_n}) \quad \text{by shift/reduce}. \\
&\xrightarrow[R]{} l(\alpha_1)\alpha_2 \cdots \alpha_n \\
B &\xrightarrow[R]{*} l(\alpha_1)\alpha_2 \cdots \alpha_n
\end{aligned}$$

In both cases, the critical pair is confluent.

Case i.3 We superpose two T-rules on one letter, say the former with n, the latter with m components $1<m\leq n$. With $l(\overline{\alpha_n})=f(\alpha'_1)$ and $f(\alpha'_1)>s(\alpha'_1)$, we have:

$$\alpha_1 \cdots \alpha_n \alpha'_1 \cdots \alpha'_m l(\overline{\alpha_m})$$
$$\longrightarrow s(\alpha_1)\alpha_1 \cdots \alpha_n \overline{\gamma'_1}\alpha'_2 \cdots \beta'_m = A \quad \text{by left reduction.}$$
$$\longrightarrow \alpha_1 \cdots \alpha_n s(\alpha'_1)\alpha'_1 \cdots \alpha'_m \quad \text{by right reduction.}$$
$$= \alpha_1 \cdots \alpha_n \overline{\alpha'_1} l(\alpha'_1)\alpha'_2 \cdots \alpha'_m = B$$

As for i.2, we have a first case when $f(\overline{\alpha'_1}) \neq l(\alpha_n)$. Then C holds for α_n and $\overline{\alpha'_1}$:

$$B \xrightarrow[T]{} s(\alpha_1)\alpha_1 \cdots \alpha_n \overline{\alpha'_1}\alpha'_2 \cdots \alpha'_m \quad \text{by induction hypothesis.}$$
$$= s(\alpha_1)\alpha_1 \cdots \alpha_n \overline{\gamma'_1} l(\overline{\alpha'_1})\alpha'_2 \cdots \alpha'_m$$
$$\xrightarrow[S]{*} s(\alpha_1)\alpha_1 \cdots \alpha_n \overline{\gamma'_1}\alpha'_2 \cdots \alpha'_m l(\alpha'_m) = A \quad \text{by shift/reduce.}$$

When $f(\overline{\alpha'_1}) = l(\alpha_n)$, we have $\alpha_n \overline{\alpha'_1} \xrightarrow[RS]{*} 1$ and $\alpha_n \overline{\gamma'_1} \xrightarrow[RS]{*} l(\gamma'_1) = l(\overline{\alpha'_1})$. Therefore,

$$A \xrightarrow[RS]{*} s(\alpha_1)\alpha_1 \cdots \alpha_{n-1}\alpha'_2 \cdots \alpha'_m l(\overline{\alpha'_m}) l(\overline{\alpha'_m}) \quad \text{by shift/reduce.}$$
$$\xrightarrow[R]{} s(\alpha_1)\alpha_1 \cdots \alpha_{n-1}\alpha'_2 \cdots \alpha'_m$$
$$B \xrightarrow[RS]{*} \alpha_1 \cdots \alpha_{n-1} l(\alpha'_1)\alpha'_2 \cdots \alpha'_m$$
$$= \alpha_1 \cdots \alpha_{n-1} l(\overline{\alpha_{n-1}})\alpha'_2 \cdots \alpha'_m$$
$$\xrightarrow[T]{} s(\alpha_1)\alpha_1 \cdots \alpha_{n-1}\alpha'_2 \cdots \alpha'_m \quad \text{by induction hypothesis.}$$

And the critical pair is confluent in both cases.

Case ii.1 The R-rule appears on the right of the least common instance.

$$\alpha_1 \cdots \alpha_n l(\overline{\alpha_n}) l(\overline{\alpha_n}) \longrightarrow s(\alpha_1)\alpha_1 \cdots \alpha_n l(\overline{\alpha_n}) \quad \text{by left reduction.}$$
$$\longrightarrow s(\alpha_1)s(\alpha_1)\alpha_1 \cdots \alpha_n \quad \text{yet by left reduction.}$$
$$\xrightarrow[R]{} \alpha_1 \cdots \alpha_n$$
$$\longrightarrow \alpha_1 \cdots \alpha_n \quad \text{by right reduction.}$$

Case ii.2 The R-rule appears on the left hand side.

$$f(\alpha_1)\alpha_1 \cdots \alpha_n l(\overline{\alpha_n}) \longrightarrow \overline{\gamma_1}\alpha_2 \cdots \alpha_n l(\overline{\alpha_n}) \quad \text{by left reduction.}$$
$$\longrightarrow f(\alpha_1)s(\alpha_1)\alpha_1 \cdots \alpha_n \quad \text{by right reduction.}$$
$$= \beta_1 l(\alpha_1)\alpha_2 \cdots \alpha_n$$
$$\xrightarrow[S]{} \overline{\beta_1} l(\alpha_1)\alpha_2 \cdots \alpha_n \quad \text{as } f(\alpha_1) > s(\alpha_1) \text{ by } C.$$
$$\xrightarrow[R]{} \overline{\alpha_1}\alpha_2 \cdots \alpha_n \quad \text{as } l(\overline{\beta_1}) = l(\alpha_1)$$
$$= \overline{\gamma_1} l(\overline{\alpha_1})\alpha_2 \cdots \alpha_n$$
$$\xrightarrow[S]{*} \overline{\gamma_1}\alpha_2 \cdots \alpha_n l(\overline{\alpha_n}) \quad \text{by shift/reduce.}$$

The critical pair is confluent in both cases.

Case iii.1 The two rules are $\alpha_1 \cdots \beta_n \longrightarrow \overline{\beta_1} \cdots \alpha_n$ and $\overline{\beta_n} \longrightarrow \beta_n$. Let $i = l(\overline{\alpha_n})$, $j = l(\alpha_n)$ and $m = m_{ij}$. If $0 < k < m$ and k even, we have:

$$\alpha_1 \cdots \alpha_n [ij]^k$$
$$\longrightarrow s(\alpha_1)\alpha_1 \cdots \alpha_n [ji]^{k-2} j \quad \text{by left reduction.}$$
$$\xrightarrow[R]{*} s(\alpha_1)\alpha_1 \cdots \alpha_{n-1}[f(\alpha_n)s(\alpha_n)]^{m-k} \quad \text{by } k-1 \text{ reductions.}$$
$$\longrightarrow \alpha_1 \cdots \alpha_{n-1}[f(\alpha_n)s(\alpha_n)]^{k-1}\beta_n \quad \text{by right reduction.}$$

$\xrightarrow[R]{*} \alpha_1 \cdots \alpha_{n-1}[s(\alpha_n)f(\alpha_n)]^{m-k+1}$ by $k-1$ reductions.

$= \alpha_1 \cdots \alpha_{n-1}s(\alpha_n)[f(\alpha_n)s(\alpha_n)]^{m-k}$

$= \alpha_1 \cdots \alpha_{n-1}l(\overline{\alpha_{n-1}})[f(\alpha_n)s(\alpha_n)]^{m-k}$

$\xrightarrow[T]{} s(\alpha_1)\alpha_1 \cdots \alpha_{n-1}[f(\alpha_n)s(\alpha_n)]^{m-k}$ by induction hypothesis.

If $i=f(\alpha_1)$, $j=s(\alpha_1)$ and $m=m_{ij}$, the S-rule is now $\beta_1 \longrightarrow \overline{\beta_1}$. If $0<k<m$ and k even, we have:

$[ij]^k\alpha_1 \cdots \alpha_n l(\overline{\alpha_n})$

$\longrightarrow \overline{\beta_1}[l(\alpha_1)l(\overline{\alpha_1})]^{k-1}\alpha_2 \cdots \alpha_n l(\overline{\alpha_n})$ by left reduction.

$\xrightarrow[R]{*} [ji]^{m-k} l(\overline{\alpha_1})\alpha_2 \cdots \alpha_n l(\overline{\alpha_n})$ by $k-1$ reductions.

$\xrightarrow[S]{*} [ji]^{m-k}\alpha_2 \cdots \alpha_n l(\overline{\alpha_n})l(\overline{\alpha_n})$ by shift/reduce.

$\xrightarrow[R]{} [ji]^{m-k}\alpha_2 \cdots \alpha_n$

$\longrightarrow [ij]^k s(\alpha_1)\alpha_1 \cdots \alpha_n$ by right reduction.

$\xrightarrow[R]{*} [ji]^{m-k}\alpha_2 \cdots \alpha_n$ by k reductions.

We have confluence in both cases.

Case iii.2 We superpose on more than one generator two T-rules with left members $\alpha_1 \cdots \beta_n$ and $\alpha'_1 \cdots \beta'_m$ with $1<m\leq n$. By the first inequality C, the matching word is a proper subword of α_n. Let $i=l(\alpha_n)$, $j=l(\overline{\alpha_n})$ and $m=m_{ij}$. We have $f(\alpha'_1)=s(\alpha_n)$, $s(\alpha'_1)=f(\alpha_n)$ and $l(\alpha'_1)=l(\overline{\alpha_n})$.

$\alpha_1 \cdots \alpha_n j[ij]^k\alpha'_2 \cdots \beta_m$ where $0\leq k<m-2$ and k even.

$\longrightarrow s(\alpha_1)\alpha_1 \cdots \alpha_n[ij]^k\alpha'_2 \cdots \beta_m$ by left reduction.

$\xrightarrow[R]{*} s(\alpha_1)\alpha_1 \cdots \alpha_{n-1}[f(\alpha_n)s(\alpha_n)]^{m-1-k}\alpha'_2 \cdots \beta'_m$ by k reductions.

$= s(\alpha_1)\alpha_1 \cdots \alpha_{n-1}[f(\alpha_n)s(\alpha_n)]^{m-k}l(\overline{\alpha'_1})\alpha'_2 \cdots \beta'_m$

$\xrightarrow[RS]{*} s(\alpha_1)\alpha_1 \cdots \alpha_{n-1}[f(\alpha_n)s(\alpha_n)]^{m-k}\alpha'_2 \cdots \alpha'_m$ by shift/reduce.

$\longrightarrow \alpha_1 \cdots \alpha_{n-1}[f(\alpha_n)s(\alpha_n)]^{k+1}s(\alpha'_1)\alpha'_1 \cdots \alpha'_m$ by right reduction.

$\xrightarrow[R]{*} \alpha_1 \cdots \alpha_{n-1}[f(\alpha'_1)s(\alpha'_1)]^{m-k-1}\alpha'_2 \cdots \alpha'_m$ by $k+1$ reductions.

$\xrightarrow[T]{} s(\alpha_1)\alpha_1 \cdots \alpha_{n-1}[f(\alpha_n)s(\alpha_n)]^{m-k}\alpha'_2 \cdots \alpha'_m$

The last step follows from $f(\alpha'_1)=s(\alpha_n)=l(\overline{\alpha_{n-1}})$ and induction hypothesis. It remains to check the normal pairs.

Case iv.1 The rule is $\alpha_1 \cdots \beta_n \longrightarrow \overline{\beta_1} \cdots \alpha_n$. A first normal pair $(\alpha_1 \cdots \alpha_n\ ,\ s(\alpha_1)\alpha_1 \cdots \alpha_n l(\overline{\alpha_n}))$ is trivially confluent. The second one is $(\gamma_1\alpha_2 \cdots \beta_n\ ,\ f(\alpha_1)s(\alpha_1)\alpha_1 \cdots \alpha_n)$. The last word is equal to:

$\beta_1 l(\alpha_1)\alpha_2 \cdots \alpha_n \xrightarrow[S]{} \overline{\beta_1}l(\alpha_1)\alpha_2 \cdots \alpha_n$ as $f(\alpha_1)>s(\alpha_1)$

$= \overline{\alpha_1}l(\overline{\beta_1})l(\alpha_1)\alpha_2 \cdots \alpha_n$

$\xrightarrow[R]{} \overline{\alpha_1}\alpha_2 \cdots \alpha_n$ as $l(\overline{\beta_1})=l(\alpha_1)$

$\xrightarrow[S]{*} \overline{\gamma_1}\alpha_2 \cdots \alpha_n l(\overline{\alpha_n})$ by shift/reduce.

Case iv.2 The last normal pair takes the inverses. The critical pair (A,B) is $(l(\overline{\alpha_n})\alpha_n^{-1} \cdots \alpha_1^{-1},\ \alpha_n^{-1} \cdots \alpha_1^{-1}s(\alpha_1))$. Note that if m_α is even then $\alpha^{-1}=\alpha$, otherwise $\alpha^{-1}=\overline{\alpha}$. If m_{α_n} is even, then:

$$A \longrightarrow \beta_n \alpha_{n-1}^{-1} \cdots \alpha_1^{-1} \quad \text{by } \overline{\beta_n} \longrightarrow \beta_n.$$
$$= \alpha_n l(\overline{\alpha_{n-1}})\alpha_{n-1}^{-1} \cdots \alpha_1^{-1} \quad \text{as } l(\overline{\alpha_n})=s(\alpha_n)=l(\overline{\alpha_{n-1}}).$$

The same reduction sequence applies while m_k is even. Thus there is confluence if all m are even, otherwise:

$$A \xrightarrow[S]{*} \alpha_n \cdots \alpha_{k+1} l(\overline{\alpha_k})\alpha_k^{-1} \cdots \alpha_1^{-1} \quad \text{with } m_{\alpha_k} \text{odd}.$$

But $B=\alpha_n \cdots \alpha_{k+1}\alpha_k^{-1} \cdots \alpha_1^{-1}s(\alpha_1)$ and the normal pair $(l(\overline{\alpha_k})\alpha_k^{-1} \cdots \alpha_1^{-1}, \alpha_k^{-1} \cdots \alpha_1^{-1}s(\alpha_1))$ is resolved by induction hypothesis. If m_{α_n} is odd, then A is irreducible. We have two subcases. If m_{α_1} is even, then

$$B \longrightarrow \alpha_n^{-1} \cdots \alpha_2^{-1}s(\alpha_1)\alpha_1 \quad \text{by } \beta_1 \longrightarrow \overline{\beta_1}.$$
$$= \alpha_n^{-1} \cdots \alpha_2^{-1}s(\alpha_2)\alpha_1 \quad \text{as } s(\alpha_1)=l(\overline{\alpha_1})=s(\alpha_2).$$

And, starting at α_2^{-1}, this reduction is applied while m_α is odd. Either all remaining m's are odd, in which case there is confluence, or one m_{α_k} is even. Then a T-rule applies whose last component is α_k, the other ones being a sequence of even α's ending with an odd one which surely exists as m_{α_n} is odd. This T-reduction does not halt this new shift/reduce process, it restarts according to the parity of the m_α's. Thus under a loop of S-reductions followed by T-reductions the pair conflues.
If m_{α_1} is odd, the same loop appears starting with a T-reduction. This concludes the proof ■

Let us see an example (when we give a complete set of rules, we shall omit the first ones from R_l). The group is defined on three generators a, b and c with $m_{ab}=4$, $m_{bc}=5$ and $m_{ca}=6$. We give two complete presentations, the first one is defined by the ordering $b>a>c$:

$$G1 \left\{ \begin{array}{rcl} baba & \longrightarrow & abab \\ bcbcb & \longrightarrow & cbcbc \\ acacac & \longrightarrow & cacaca \\ bcbcabab & \longrightarrow & cbcbcaba \\ babcacaca & \longrightarrow & ababcacac \\ bcbcabacbcbc & \longrightarrow & cbcbcabacbcb \end{array} \right.$$

Each rule is a T one. The confluence of the system has been mechanically checked on a DPS8-Multics by the system Al-Zebra [6] (as for all examples in the paper). The complete system has twelve rules. A smaller system is associated with the ordering $a>c>b$:

$$G2 \left\{ \begin{array}{rcl} abab & \longrightarrow & baba \\ cbcbc & \longrightarrow & bcbcb \\ acacac & \longrightarrow & cacaca \\ acacabcbcb & \longrightarrow & cacacabcbc \end{array} \right.$$

Thus, the number of rules depends on the ordering. However, this number does not matter if all rules fall under a single parametrized one. The set of rules may be infinite. Here is an example:

$$G3 \left\{ \begin{array}{rcl} dcd & \longrightarrow & cdc \\ dbdb & \longrightarrow & bdbd \\ dada & \longrightarrow & adad \end{array} \right.$$

The completion of this set creates infinitely many rules of type $dcbdb\,(adabdb)^n d \longrightarrow cdcbdb\,(adabdb)^n,\ n \geq 0$. Thus we have examples of infinite sets of identities defining efficient algorithms. Noteworthy, all T-rules are in Post normal form: they can be written as $Va \longrightarrow bV$, where V is $\alpha_1 \cdots \alpha_n$. Before the study of a reduction algorithm, we examine presentations with commuting pairs of generators.

2.2. Commuting pairs of generators

First, let M be a Coxeter matrix with infinite m_{ij}'s, then the meta-rule T always gives the complete system, and theorem 3 is valid for M with the convention that no component α exists for i and j. The meta-rule is puzzled only when some entries of M are equal to 2, i.e. $ij = ji$, the generators commute. This case implies that instances of the meta-rule are S-reducible. With the following *complete* system:

$$G4 \begin{cases} ad \longrightarrow da \\ bd \longrightarrow db \\ ca \longrightarrow ac \\ cd \longrightarrow dc \\ cbc \longrightarrow bcb \\ cbac \longrightarrow bcba \\ cbabcb \longrightarrow bcbabc \end{cases}$$

The T-rule $cbadc \longrightarrow bcbad$ is never created as its members are confluent under the commutativity rules and the T-rule $cbac \longrightarrow bcba$. Critical pairs are reducible with S-rules, then with arbitrary T-ones. Moreover, some T-rules are partially reduced by the commutativity laws (cf. in last section B_n complete presentations). As final drawback, when a Coxeter matrix has infinite coefficients and others equal to 2, new kinds of rules appear, with:

$$G5 \begin{cases} ca \longrightarrow ac \\ cb \longrightarrow bc \\ dad \longrightarrow ada \\ dbd \longrightarrow bdb \\ dcd \longrightarrow cdc \end{cases}$$

the completion procedure generates infinitely many rules $dxcd[yx]^n c \longrightarrow xdxcd[yx]^n,\ n \geq 0$ where $\{x,y\} = \{a,b\}$. To moderate these negative results, the last section presents some complete systems of Coxeter groups with commuting pairs of generators.

3. THE REDUCTION ALGORITHM

The simplest reduction algorithm iterates the search of a left member, and the substitution of right members. This is of little practical interest. We begin by some remarks on rules and overlapping reductions.

Our goal is a reduction algorithm without backward search in a word already scanned and reduced. After a reduction, what are the possible ones overlapping the new right member ? We first restrict our attention to T-rules:

$$\begin{aligned} W &\xrightarrow{*} v\alpha_1 \cdots \alpha_{n-1}\beta_n w \\ &\longrightarrow v\overline{\beta_1}\alpha_2 \cdots \alpha_n w \end{aligned}$$

Any reduction of a v suffix also reduces at most the subword $\overline{\beta_1}$ by the condition C. If it reduces a $\overline{\beta_1}$ prefix of at least two letters, then the same condition C between $\overline{\beta_1}$ and the v suffix implies that $v\alpha_1$ is also reducible. In order to avoid backtracking, the reduction strategy is leftmost, keeping the word v irreducible. If the $\overline{\beta_1}$ prefix is a single generator, reductions may occur, with example G1, we have:

acaca bcbcabab —>*acaca* **c** *bcbcaba* =*acaca* **c** *bcbcaba* —>*cacacabcbcaba* (E1)

Thus, the algorithm must update a stack of old left redexes. These prefixes are kept with their occurrence number in the reduced part of the input word. Note that all such prefixes must be stored: we can reach a word vpv_1v_2sw where v_2 is the reverse of v_1, vpv_1 is irreducible and ps is a left member. Then by S,R-reductions, we get the word $vpsw$.

On the right side of the right member, inequalities C imply only one possible reduction: a suffix of the last component α_n is a prefix of the first component of the next redex. But before the reduction we had the configuration $v\alpha_1 \cdots \alpha_n l(\overline{\alpha_n}) l(\overline{\alpha_n}) w'$. Also, to overcome such overlapping reductions, the algorithm will R-reduce on the right of a T-redex before the T-reduction. Therefore, we may restart the search on the last generator of the right member. With the same example G1:

bcbcabab cacac —>*cbcbcab* **a** *cacac* =*cbcbcab* **a***cacac* —>*bcbcabcacaca* (E2)

The two facts that 1) the new generator (c in E1) can only be the last one of a new redex just before the old one and 2) we can skip until the last generator of the right member (a in E2) are the basic points of the reduction algorithm.

Let us now look more closely to the possible R-reductions following a T-one. On the left, we claim that at most one R-reduction can occur. Otherwise, the T-redex would not be the leftmost one as $f(\alpha_1)s(\alpha_1)\alpha_1 \longrightarrow \overline{\beta_1}l(\alpha_1)$. With example G1:

$$bc\ bcbcabab \longrightarrow bc\ cbcbcaba \overset{*}{\longrightarrow} cbcaba$$

But $bc\ bcbcabab = bcbcb\ cabab$, and the redex $bcbcabab$ is not the leftmost one. On the right hand of a right member, we may of course have several R-reductions. Here we may observe that the leftmost strategy is also more efficient than the rightmost one:

$$bcbcabab\ a \longrightarrow cbcbcaba\ a \longrightarrow cbcbcab$$

While $bcbca\ baba \longrightarrow bcbca\ abab \longrightarrow bcbcbab \longrightarrow cbcbcab$

The rightmost strategy induces a shift/reduce process wich makes n+1 reductions while the leftmost one always produce two reductions, n the number of components in the leftmost redex.

It remains to examine the consequences of R-reductions. On another R-reduction, they are taken in account by a loop deleting the common generators at top and bottom of the unscanned and reduced part of the input word. On a T-rule, a R-reduction may increase the last redex prefix at top of the redex stack. Thus a sequence of R-reductions must be closed by an update. This update splits in two operations: the removal of prefixes deleted by R-reductions, and possibly a pop operation on the stack, so that its top becomes the current prefix.

These two observations outline the global strategy of an efficient reduction algorithm based on a leftmost strategy. Let us present more formally the main iteration of the algorithm. This loop updates four variables: v the reduced part of the initial word, w the remaining input, r the current T-redex prefix, and s, the stack: list of redex prefixes together with their occurrence in v. Entering the loop, the word is equal to vrw, where $r=\alpha_1 \cdots \alpha_k$, C being satisfied. Building r needs a function ***component*** which recognizes a component in the word w, i.e. $w=\alpha k w'$, where $k \neq l(\alpha)$. Also a boolean function ***link*** returns true if condition C between the last component α_k of r and α is satisfied, observe that this function uses the assumption $m_{ij} \neq 2$. A procedure ***apply*** applies the rule whose left member has just been recognized, this function possibly pops the stack as a R-rule may appear on the left of the redex right member. The update of r, v and the stack s after a R-reduction is performed by a procedure ***update***. Finally, ***plast*** returns the last generator of $\bar{\alpha}$, and, given two generators a,b, ***alpha*** computes α_{ab}. A list is noted [a;b;c], the dot . and the at @ are the list cons and append, ***last*** and ***tail*** are usual list functions.

Main loop of the reduction algorithm

```
loop {
While w=aaw' do { w:=w' } ;
While w=abbw' do { w:=aw' } ;
If a=last(v@r) Then { w:=tail(w); update(v,r,s) }
Else {
(bool, w', c, d):=component(a,b,(tail (tail w)));
If bool Then {                (w starts with a component)
  If a>b Then { If c=plast(a,b) Then { apply([],a,b,(d.w'),(v@r)) }
                Else { v:=v@r; push(s,r); r:=alpha(a b) } }
  Else { If r=[] Then { v:=v@alpha(a,b) }
         Else { If c=plast(a,b) and link(a,b,r) Then { apply(r,a,b,(d.w'),v) }
                Else { r:=r@alpha(a,b) } } } }
Else { v:=v@[a;b]; w:=w'; push(s,r); r:=[] } } }
```

4. SOME EXAMPLES

In all examples, the set R_I is omitted. We consider finite Coxeter groups first described by H.S.M Coxeter [2]. The notations are taken from [1]. The completion of a finite group always halts [6]. The finite Coxeter groups whose matrix entries are equal to 1,2,3,4 or 6 are called crystallographic groups. However we failed to complete the following crystallographic groups: E_n, n=6,7,8 and the family D_n.

Groups H_4, H_3 and $I_2(n)$.

$$H_4 \left\{ \begin{array}{rcl} dcd & \longrightarrow & cdc \\ cbc & \longrightarrow & bcb \\ cbac & \longrightarrow & bcba \\ babab & \longrightarrow & ababa \\ cbabcb & \longrightarrow & bcbabc \\ cbabacbaba & \longrightarrow & bcbabacbab \end{array} \right. \qquad H_3 \left\{ \begin{array}{rcl} cbc & \longrightarrow & bcb \\ cbac & \longrightarrow & bcba \\ babab & \longrightarrow & ababa \\ cbabcb & \longrightarrow & bcbabc \\ cbabacbaba & \longrightarrow & bcbabacbab \end{array} \right.$$

These two groups are not crystallographic, nor are the dihedral groups $I_2(n)$, n>4, except $I_2(6)$. These groups are generated by two plane reflections through lines whose angle is $\frac{2\pi}{n}$, their complete presentation is the simpler one, all critical pairs are solved by symmetrization, the complete set is $R_I \cup S_I$. The remaining finite groups are crystallographic.

Groups F_4, E_6, E_7 and E_8.

$$F_4 \left\{ \begin{array}{r} bab \longrightarrow aba \\ dcd \longrightarrow cdc \\ cbcb \longrightarrow bcbc \\ cbacba \longrightarrow bcbacb \end{array} \right.$$

The Knuth-Bendix procedure failed to complete the three groups E_n, n=6,7,8 while generating hundred rules such as, for E_6:

$$\begin{array}{rcl} fcbadcbedfcbadc & \longrightarrow & cfcbadcbedfcbad \\ fcbadcbedcfcbdcf & \longrightarrow & cfcbadcbedcfcbdc \end{array}$$

As these groups possess many matrix entries equal to 2, we do not have an efficient reduction algorithm for these three groups.

Groups A_n, B_{n+1} and D_{n+3}, n≥1.

These three families define the infinite families of crystallographic groups. The first family is the family of symmetric groups. Their complete presentation, when presented as Coxeter groups, was given in [6]:

$$A_n \left\{ \begin{array}{rcll} R_i & = & (i\ i+1) & i=1,\ldots,n-1 \\ R_i^2 & = & 1 & i=1,\ldots,n-1 \\ R_i R_j & = & R_j R_i & i \le j-2 \\ (R_i R_{i+1})^3 & = & 1 & 1 \le i \le n-2 \end{array} \right.$$

Despite commuting pairs of generators, we have only T-rules:

$$A_n \left\{ \begin{array}{rcll} R_i R_j & \longrightarrow & R_j R_i & j \le i-2 \\ R_i R_{i-1} \cdots R_j R_i & \longrightarrow & R_{i-1} R_i R_{i-1} \cdots R_j & j<i \end{array} \right.$$

The groups B_n also possess a fair complete presentation. Their Coxeter matrix is

$$\begin{bmatrix} 1 & 3 & & & & \\ 3 & 1 & \ddots & & \text{\Large 2} & \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & 1 & 3 & 2 \\ & \text{\Large 2} & & 3 & 1 & 4 \\ & & & 2 & 4 & 1 \end{bmatrix}$$

The complete system includes the rules R_I, the rules of commutativity, and the following T-rules for B_n:

$$\left\{ \begin{array}{ll} a_i a_{i-1} \cdots a_{i-k} a_i \longrightarrow a_{i-1} a_i a_{i-1} \cdots a_{i-k} & n>i>k>0 \\ (a_n a_{n-1} \cdots a_{n-k})^2 \longrightarrow a_{n-1} a_n a_{n-1} \cdots a_{n-k} a_n a_{n-1} \cdots a_{n-k+1} & n>k>0 \end{array} \right.$$

The last family D_n however does not possess an easily described complete system. Their Coxeter matrix is the previous one where the last row and column are replaced by [2 2 ... 2 3 2 1]. Here are the T-rules except the commutative ones for D_4:

$$\begin{array}{rcl} dbd & \longrightarrow & bdb \\ cbc & \longrightarrow & bcb \\ bab & \longrightarrow & aba \end{array}$$

cbac	—>	*bcba*
dbcb	—>	*cdbc*
dbad	—>	*bdba*
dbcd	—>	*bdbc*
dbacd	—>	*bdbac*
dbacba	—>	*cdbacb*
dbacbdb	—>	*bdbacbd*

In [8], R.H. Gilman proposes a procedure which is basically the completion procedure described here. In particular, lemma 1.(3) of [8] defines the superposition between rules. The main difference with the present completion lies in the computation of normal pairs giving a completion procedure well suited for groups.

References

1. Bourbaki N., *Groupes et algèbres de Lie*, Ch. 4,5,6. Actu. Sci. Ind., Hermann, Paris. (1978).
2. Coxeter H.S.M., *The complete enumeration of finite groups of the form* $R_i^2=(R_iR_j)^{k_{ij}}=1$., J. Lond. Math. Soc.,10, pp 21-25. (1935).
3. Coxeter H.S.M. and Moser W.O.J., *Generators and Relations for Discrete Groups*, Springer-Verlag (1972).
4. Knuth D. and Bendix P., *Simple Word Problems in Universal Algebras*, In: Computational Problems in Abstract Algebra Ed. Leech J., Pergamon Press, pp. 263-297. (1970).
5. Huet G., *A Complete Proof of Correctness of the Knuth-Bendix Completion Algorithm*, J. Comp. Sys. Sci. 23,1, pp. 11-21 (Aug. 1981).
6. Le Chenadec Ph., *Formes canoniques dans les algèbres finiment présentées*, Thèse de 3ème cycle, Univ. d'Orsay (June 1983).
7. Tits J., *Le Problème des mots dans les groupes de Coxeter*, In: Sympos. Math. Rome 1967/68, pp. 175-185. Lond. Academic Press (1969).
8. Gilman R.H., *Presentations of Groups and Monoids*, J. Alg. 57, pp. 544-554. (1979).

Imprimé en France

par

l'Institut National de Recherche en Informatique et en Automatique